BSTZ No. 042390.P9144
Express Mail No. EL466331508US

UNITED STATES PATENT APPLICATION

FOR

A SYSTEM AND METHOD FOR VERIFYING THE INTEGRITY OF STORED INFORMATION WITHIN AN ELECTRONIC DEVICE

Inventor(s):

Robert P. Hale
Andrew J. Fish

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
12400 Wilshire Blvd., Suite 700
Los Angeles, California 90025
(714) 557-3800

A SYSTEM AND METHOD FOR VERIFYING THE INTEGRITY OF STORED INFORMATION WITHIN AN ELECTRONIC DEVICE

1. Field

The present invention relates to the field of data security. More particularly, this invention relates to a scheme for verifying the integrity of stored information loaded within an electronic device.

2. General Background

Many electronic devices include a set of semi-permanently stored instructions referred to as firmware. For instance, computers include a type of firmware referred to as the basic input/output system (BIOS). Being executed by a processor of the computer, the BIOS is coded to perform various functions. For example, during a pre-boot cycle at power-up, the BIOS controls the initialization of the computer as well as the initialization of various hardware peripherals. Normally provided by a single vendor, the BIOS is loaded into pre-boot space of a non-volatile memory such as a read-only memory (ROM) component or a flash memory component during manufacture of the computer.

Recently, however, it has become desirable to store more sophisticated routines and data in the pre-boot space of the non-volatile memory. As an example, in recent efforts to protect against software viruses and malicious corruption of the BIOS, an image of the BIOS code may be digitally signed to produce a digital signature. Prior to execution of the BIOS, the digital signature may be used to determine whether the BIOS has been modified. This provides much needed virus protection.

Well known in the art, a digital signature is digital data signed using a private key of its signatory. Similar to encryption, the "signing process" may be accomplished using any of a number of software algorithms such as a Rivert Shamir and Adleman (RSA) algorithm or the Digital Signature Algorithm (DSA) as set forth in a Federal Information Processing Standards publication 186 entitled "Digital Signature Standard" (May 19, 1994). Normally, the digital data is placed

in an encoded form (referred to as the "hash value"), achieved by performing a one-way hash operation on the original digital data, prior to signing the hash value. The term "one-way" indicates that there does not readily exist an inverse operation or function to recover any discernible portion of the digital data from the hash value.

Recently, the computer industry has made efforts to develop BIOS as a collection of software modules produced by different vendors rather than a piece of monolithic code produced by a single vendor. It is likely that the code of the BIOS modules would be configured as "execute-in-place" modules because this code would be executed before the availability of system random access memory (RAM). Also, it is likely that relocation would be used to properly load the BIOS modules within the non-volatile memory because it would be too difficult for all of the BIOS vendors to agree on the specific addressing scheme beforehand.

As commonly known in the industry, "relocation" is a process by which addresses within each BIOS module are adjusted based on the particular address location in memory allotted for the BIOS module (referred to as the "base address"). Thus, software routines within a BIOS module are usually coded with relative offsets from a base address that has not yet been assigned. During relocation, the addresses of various software routines within the BIOS module would be adjusted by adding the base address to each of the relative offsets.

Unfortunately, if relocation is performed on the execute-in-place BIOS modules, any digital signatures associated with the images of the BIOS modules would be ineffective because any data integrity analysis using the digital signatures would indicate that the BIOS module has been modified. Hence, it is virtually impossible to determine whether modification of the BIOS module was unauthorized or merely due to the relocation operation. Thus, it would be desirable to develop an integrity verification mechanism that improves the effectiveness of digital signatures in detecting unauthorized modifications to the BIOS module while still allowing the image to undergo relocation.

Moreover, when BIOS is developed as a collection of digitally signed BIOS modules produced by different vendors, in certain situations, it may be

desirable to dynamically link these digitally signed modules. In particular, one BIOS module may be configured to make a call for a function coded in another BIOS module. However, in order to dynamically link the BIOS modules together, it would require modification of at least one BIOS module, which would invalidate any digital signature associated with the image of that BIOS module. Thus, the original digital signatures would not be effective to identifying unauthorized modification of the module. Thus, an integrity verification mechanism that overcomes this problem would be desirable.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is an illustrative block diagram of a collection of software modules for loading as firmware into an electronic device.

Figure 2 is an illustrative block diagram of an embodiment of an electronic device utilizing the present invention.

Figure 3 is a block diagram of a first illustrative embodiment of the contents of the non-volatile memory component of Figure 2 that are collectively used to verify the integrity of relocated, post-relocation images using digital signatures.

Figure 4 is a block diagram of a second illustrative embodiment of the contents of the non-volatile memory component of Figure 2.

Figure 5 is a flowchart of the operations for verifying the integrity of stored information, such as a post-relocation image shown in Figures 3 and 4.

Figure 6 is a block diagram of a second illustrative embodiment of the present invention featuring a plurality of digitally signed images are dynamically linked together through one or more Bound & Relocated Import Tables (BRITs).

Figure 7 is a flowchart of the operations for generating a Bound and Relocated Import Table (BRIT).

Figure 8 is a flowchart of the operations for verifying the Bound and Relocated Import Table (BRIT) of Figure 7.

DETAILED DESCRIPTION OF THE INVENTION

Herein, certain embodiments of the invention are described for verifying the integrity of information that is stored within an electronic device during pre-boot operations. In general, the stored information may include, for example, a digitally signed image that includes a post-relocation image of a software module or is dynamically linked with another digitally signed image.

In the following description, certain terminology is used to discuss features of the present invention. A "software module" comprises a set of instructions that perform a particular function. For example, the software module may feature instructions that are executed during a pre-boot cycle in order to initialize an electronic device. A replication of a binary representation of the instructions associated with the software module is referred to as an "image". Different types of images can be used to represent different formatting stages. For instance, a "pre-relocation image" is a binary representation of the software module prior to conducting a relocation operation thereon. A "post-relocation image" is a binary representation of the module after relocation.

Furthermore, an "electronic device" is a combination of electronic hardware and software that collectively operates to perform one or more specific functions. Examples of an electronic device include a computer (e.g., a laptop, desktop, hand-held, server, mainframe, etc.), a component of the computer (e.g., a serial port), a cellular telephone, a set-top box (cable box, network computer, satellite television receiver, etc.), a network appliance and the like. A "link" is broadly defined as one or more information-carrying mediums to establish a communication pathway, including physical medium (e.g., electrical wire, optical fiber, cable, bus traces, etc.) or wireless medium (e.g., air in combination with wireless signaling technology).

Briefly, one integrity verification mechanism involves the configuration of a digitally signed image to include relocation information, a post-relocation image and a digital signature. The "relocation information" is a series of relative offsets from a base address. These offsets are generated after the stored information (e.g.,

an image of a software module) is compiled and placed into an executable format such as an MS-DOS® "EXE" format (MS-DOS is a registered trademark of Microsoft Corporation of Redmond, Washington). The offsets are converted to appropriate addresses during relocation when the base address, namely the storing address at which the image of the software module is stored and retrieved for execution, is determined. Thus, the post-relocation image differs from a pre-relocation image. The digital signature, however, is based on the pre-relocation image.

Another second integrity verification mechanism involves the inclusion of an import table and an export table within each digitally signed image. These tables allow functions within different digitally signed images to be dynamically bound together via a Bound & Relocated Import Table (BRIT). The BRIT resides outside the digitally signed image. Both of the integrity verification mechanisms may be performed by hardware or a software program embedded in a processor (described below) or simply executable by the processor.

Referring to Figure 1, an illustrative block diagram of a collection of "N" software modules ready for loading as firmware 100 into an electronic device is shown. Herein, each software module 110_N ($N \geq 1$) includes a header 120_N and an image 130_N for a particular software segment of the firmware 100. Prior to loading the software modules as firmware into a non-volatile memory as described below, each image 130_N is digitally signed by a signatory to produce a digital signature 140_N. The signatories may differ between each module or multiple modules may share the same signatory. A "signatory" may include any person or entity in a position of trust to guarantee or sponsor the digital signature (e.g., a bank, governmental entity, trade association, original equipment manufacturer, vendor, etc.).

Referring now to Figure 2, an illustrative block diagram of an embodiment of an electronic device is shown. For this embodiment, the electronic device 200 comprises a chipset 210 coupled to a processor 220 and a memory 230 through a first bus 240 and a second bus 250, respectively. In addition, chipset 210 is coupled to a third bus 260 that provides a pathway to one or more system

resources 270. Herein, the third bus 260 is represented as an input/output (I/O) bus (e.g., Peripheral Component Interconnect "PCI" bus); however, any other type of bus architecture may be used, including such bus architectures as Industry Standard Architecture (ISA), Extended ISA (EISA), Universal Serial Bus (USB) and the like. Herein, the third bus 260 is shown as a single bus, but it is contemplated that the third bus 260 may include multiple buses coupled together through bridge circuitry.

As shown, the system resources 270 would be coupled to at least one of the multiple buses. The system resources 270 comprise a communication device 280 and a non-volatile memory component 290. Communication device 280 is configured to establish communications with another electronic device over a communication link 285. Examples of communication device 280 include a network interface card, a modem card or an external modem. The non-volatile memory component 290 includes firmware that features digitally signed images of one or more software modules. In one embodiment, one or more of these software modules may form a Basic Input/Output System (BIOS) code of the electronic device 200. Examples of the non-volatile memory component 290 include a programmable, non-volatile memory such as flash memory, battery-backed random access memory (RAM), read only memory (ROM), erasable programmable ROM (EPROM), electrically erasable PROM (EEPROM), or any other type of memory appropriate for storing the module(s).

Referring to Figure 3, a block diagram of a first illustrative embodiment of the loading and storage contents of the non-volatile memory component 290 of Figure 2 is shown. The non-volatile memory component 290 is loaded with one or more digitally signed images 300, which collectively act as firmware. With respect to this embodiment, a digitally signed image 300 includes relocation information 310, a pre-relocation image 320 and a digital signature 330. The positioning of the elements forming any image is a design choice.

The relocation information 310 includes relative offsets 315 for certain routines within the pre-relocation image 310. Normally, the offsets 315 are generated when the software module associated with the digitally signed image is

compiled. The offsets 315 are used for properly addressing segments of information within the software module during relocation once the starting location of the image 300, referred to as base address "B_ADDR," is determined. The relocation is conducted by a symmetrical relocation function that allows the relocated information to be undone for data integrity verification using the digital signature 330.

Herein, during relocation, the pre-relocation image 320 is converted (relocated) to a post-relocation image 340 is based on the pre-relocation image 320 of the image 300 during loading. Namely, the pre-relocation image 320 is relocated for retrieval from the base address (B_ADDR) allotted to the image 300. In essence, the relocation operation adds B_ADDR to the offsets 315 contained within the relocation information 310. This modifies the binary image such as the post-relocation image 340 stored in the non-volatile memory component now differs from the pre-relocation image 320 coded by the vendor.

The digital signature 330 includes at least a hash value of the pre-relocation image 320, which is digitally signed with a private key (PRK) of a signatory. Although the post-relocation image 340 now resides in the non-volatile memory component after relocation, it is appreciated that the digital signature 330 is based on the pre-relocated image 320 which is the binary form as originally produced before loading into the non-volatile memory component.

Referring to Figure 4, a block diagram of a second illustrative embodiment of the contents of the non-volatile memory component 290 is shown. The non-volatile memory component 290 contains multiple digitally signed images 410_1-410_M ("M" being a positive whole number) forming the firmware 400 (e.g., the BIOS). For instance, as an illustrative example, each digitally signed image 410_1-410_M is formed with a pre-relocation image 420_1-420_M, relocation information 430_1-430_M and a digital signature 440_1-440_M. Each digital signature 400_1-400_M is based on at least a hash value of its corresponding pre-relocation image 420_1-420_M and is digitally signed with a private key (PRK) of one or more signatories. Upon being loaded with the digitally signed images 410_1-410_M, the non-volatile memory component 290 undergoes a relocation operation which modifies the stored images

from the pre-relocation images 420_1-420_M to a post-relocation images 450_1-450_M, respectively.

Referring now to Figure 5, a flowchart of the operations for verifying the integrity of stored information, such as a post-relocation image of Figures 3 and 4, is shown. For integrity verification, the post-relocation image of a digitally signed image is reconverted to a pre-relocation image (block 500). This is accomplished using the relocation information contained in the digitally signed image. In particular, one or more arithmetic operations are performed on each offset; namely, as an example, the base address associated with memory of the non-volatile memory component is subtracted from each offset set forth in the relocation information. Thereafter, in block 510, a hash operation is performed on the reconverted, pre-relocation image to produce a hash value (referred to as the "reconverted hash value").

The digital signature of the digitally signed image is accessed and the hash value of the digital signature is recovered (block 520). This may be accomplished by running the digitally signed image through the digital signature algorithm being provided with a public key of the signatory for decode purposes. Thereafter, the recovered hash value is compared to the reconverted hash value (block 530). If a match is determined, the post-relocation image has been verified (block 540). Otherwise, the post-relocation image has not been verified, indicating that the image has been modified beyond such modification caused by relocation (block 550).

Figure 6 is a block diagram of a second illustrative embodiment of the present invention in which a plurality (M) of digitally signed images 600_1-600_M are dynamically linked together through one or more Bound & Relocated Import Tables (BRITs). Each BRIT corresponds to only one digitally signed image. It is contemplated that each digitally signed image 600_1-600_M may include a BRIT or only a subset of the digital signed images 600_1-600_M may be provided BRITs.

In this embodiment, a dynamic linking of two digitally signed images 600_1 and 600_M is shown. Herein, the digitally signed image 600_1 includes a BRIT 610_1, an import table 620_1, an export table 630_1, an image 640_1 based on selected

information (e.g., a software module) and a digital signature 650_1. The digital signature 650_1 is generated by conducting a one-way hash operation on the import table 620_1, the export table 630_1 and the image 640_1 to produce a resultant hash value. The resultant hash value is digitally signed by a signatory using its private key.

In general, the import table 620_1 is listing of stored information located in another digitally signed image (e.g., image 640_M) that need to be accessed for proper execution of the image 640_1. The import table 620_1 comprises a plurality of entries 625 of which at least one entry (e.g., entry 626) of the import table 620 comprises an identifier 627 and a first offset 628. Generated either internally within the electronic device or remotely by a centralized authority, the identifier 627 indicates what segment(s) of information (e.g., a function, routine, code, data, etc.) not contained within the digitally signed image 600_1, is required by the image 640_1 during execution. The identifier 627 may be represented as an alphanumeric name or a guaranteed unique identification (e.g., a sixteen-byte value). The first offset 628 is an offset pointer to an entry of the BRIT 610 that corresponds to entry 626.

The export table 630 is a listing of information contained in a digitally signed image that are available for other digitally signed images to retrieve. Entries of the export table 630_M, for example, include an identifier 635 for each segment of information included in the image 640_M and a second offset 636. The second offset 636 is equivalent to an offset from an address location of the digitally signed image 600_M to the address location of the segment of information required by image 640_1 of the digitally signed image 600_1.

As shown, the BRIT 610_1 is associated with the digitally signed image 600_1. Each entry of the BRIT 610_1 includes the identifier 627 and an address pointer 611 of the location of the segment of information. The address pointer 611 is an arithmetic combination of the starting address of the image 640_M and the second offset 636. Thus, during execution of image 600_1, a request for a segment of information referenced by the identifier 627 is routed via the BRIT 610_1 to a location within the image 640_M as represented by dashed line 660. This enables

the segment of information at that location to be accessed without modification of the image 640_M. Thus, the digital signatures 650_1 and 650_M can still be used to monitor modification of the import tables 620_1 and 620_M, export tables 630_1 and 630_M, and/or images 640_1 and 640_M.

Referring now to Figure 7, a flowchart of the operations for generating a Bound and Relocated Import Table (BRIT) of the first digitally signed image 600_1 of Figure 6 is shown. Initially, all digitally signed images within the non-volatile memory component are located (block 700). Thereafter, an import table of the first digitally signed image is located (block 710). For an initial entry of the import table, the identifier is determined and a search is conducted for a matching identifier in an export table of another digitally signed images, namely any other digitally signed image besides the first digitally signed image (blocks 720 and 730).

If the matching identifier is not located, an error is reported (blocks 740 and 750). If the matching identifier is located within a second digitally signed image, for example, the offset in the export table that corresponds to the matching identifier and resides in second digitally signed image is arithmetically combined with the starting address of the second digitally signed image (blocks 740 and 760). The combined address is loaded into an entry of the BRIT along with the identifier associated with the import table (block 770). This process continues until all entries in the import table have corresponding entries in the BRIT (block 780).

Referring to Figure 8, a flowchart of the operations for verifying the Bound and Relocated Import Table (BRIT) of Figure 7 is shown. In this embodiment, a list of all digitally signed images is generated (block 800). For each digitally signed image, verify the integrity of these digitally signed images by confirming that its corresponding import table, export table and image have not been modified (block 810). For a first digitally signed image, for example, this can be accomplished by performing a hash operation on the import table, export table and image of the first digitally signed image. This produces a resultant hash value. The resultant hash value is compared with a hash value uncovered from the digital

signature associated with the first digitally signed image. If the resultant hash value matches the recovered hash value, the import table, export table and image for the first digitally signed image have not been modified. This operation is continued for all of the remaining digitally signed images.

If the integrity of the digitally signed images cannot be verified, an error is reported (block 820). Otherwise, for the first digitally signed image, a determination is made whether the identifier in its import table matches an identifier in an export table of another digitally signed image (block 830). If no match is located, an error is reported (see block 820). If a match is located, a determination is made whether the BRIT entry corresponding to the identifier of the import table points to an address defined by the matching identifier of the export table of another digitally signed image (block 840). Since the BRIT can only point to an address defined by an export table that is contained in a digitally signed image, it can only point to trusted information. If the BRIT entry corresponding to the identifier of the import table points to an address defined by the matching identifier of the export table of another digitally signed image, the BRIT is verified (block 850). Otherwise, the BRIT is not verified (block 860).

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art.